

Technologies de l'Internet — Module TR2 : Exercices

IUT de Villetaneuse — R&T 2^{ème} année

Laure Petrucci

31 octobre 2007

1 TCP/IP — Internet

Exercice 1.1 : Analyse de trame Ethernet

Analyser la trame Ethernet suivante.

```
0000  00 04 e2 cd b8 32 00 0c f1 38 1d e7 08 00 45 00  .....2.. .8....E.
0010  00 46 00 00 40 00 40 11 b4 f1 c0 a8 02 64 c0 a8  .F..@.@. ....d..
0020  02 01 80 23 00 35 00 32 da f1 92 0d 01 00 00 01  ...#.5.2 .....
0030  00 00 00 00 00 00 01 31 01 32 03 31 36 38 03 31  ....1 .2.168.1
0040  39 32 07 69 6e 2d 61 64 64 72 04 61 72 70 61 00  92.in-ad dr.arpa.
0050  00 0c 00 01  ....
```

Exercice 1.2 : Acheminement de paquets et ARP

Deux réseaux locaux LAN_1 (d'adresse 192.6.10.0) et LAN_2 (d'adresse 194.254.173.0) sont connectés via un routeur R_1 . Un routeur R_2 , connecté au réseau local LAN_2 , permet d'accéder à un autre réseau.

La machine A , connectée au LAN_1 et d'adresse IP 192.6.10.1, reçoit de la couche TCP un message à transmettre à une machine destinataire.

Le réseau comporte aussi une machine B d'adresse IP 192.6.10.3, une machine C d'adresse 194.254.173.2, et une machine D d'adresse 220.1.1.215.

Question 1 : Faire un schéma du réseau, en plaçant LAN_1 , LAN_2 , R_1 , R_2 , A , B , C et D .

Question 2 : On suppose que la table de correspondance entre adresses IP et adresses MAC de la couche ARP est vide. Détailler les opérations ayant lieu aux niveaux 3 et 2 dans le cas où le destinataire est :

1. la machine B ;
2. la machine C ;
3. la machine D .

2 Programmation sockets UDP/TCP

Identification d'une machine hôte sur internet Une machine peut être désignée de deux façons différentes :

1. 194.254.163.24 : adresse IP notation nombres et points (ou aussi ascii)
2. 0x18a3fec2 : adresse IP binaire (ici hexa pour voir les octets)

La forme 2 correspond à la structure `in_addr` qui est définie ainsi dans `netinet/in.h` :

```
struct in_addr {
    unsigned long int s_addr;
}
```

On n'utilise pas directement un entier long pour pouvoir modifier ultérieurement la longueur de l'adresse (par exemple pour IPv6) sans changer les programmes. De ce fait, toute manipulation d'une adresse doit mentionner la longueur de cette adresse.

Il existe également une troisième forme : `lipn.univ-paris13.fr`, par exemple, qui est appelée *nom symbolique* (ou aussi *nom DNS*). L'utilisation de cette dernière est liée à l'existence d'une équivalence dans le fichier `/etc/hosts` ou à la disponibilité d'un service de noms (DNS, NIS, ...). Cette troisième forme peut souvent être utilisée à la place de la première, car elles sont toutes deux des chaînes de caractères.

Fonctions de conversion Il est possible de passer de la forme 1 à la forme 2 et inversement à l'aide de *fonctions de conversion*.

```
int inet_aton (const char *cp, struct in_addr *inp);
```

`inet_aton()` convertit l'adresse Internet `cp` (notation nombres et points) en une donnée binaire (ordre réseau), et la place dans la structure pointée par `inp`. La fonction `inet_aton()` renvoie une valeur non nulle si l'adresse est valide, et 0 sinon.

```
char *inet_ntoa (struct in_addr in);
```

`inet_ntoa()` convertit l'adresse Internet `in` (notation binaire réseau) en une chaîne de caractères dans la notation nombres et points. La chaîne est renvoyée dans un buffer alloué statiquement, qui est donc écrasé à chaque appel.

En ce qui concerne le nom symbolique, il doit être traduit par le DNS sous une des formes 1 ou 2 avant de pouvoir être utilisé par les programmes. De plus, une machine (un hôte) peut être connue sous plusieurs noms symboliques différents et avoir plusieurs adresses IP différentes. C'est pourquoi toutes les informations concernant une machine sont rassemblées dans une même structure de données `hostent` qui est définie comme suit dans `netdb.h` :

```
struct hostent {
    char *h_name;                /* Nom officiel de l'hôte */
    char **h_aliases;           /* Liste d'alias */
    int h_addrtype;             /* Type d'adresse de l'hôte */
    int h_length;               /* Longueur de l'adresse */
    struct in_addr **h_addr_list; /* Liste d'adresses */
}
#define h_addr h_addr_list[0]  /* pour compatibilité */
```

Les champs de la structure `hostent` sont :

`h_name` est le nom symbolique officiel de l'hôte.

`h_aliases` est une table, terminée par 0, d'autres noms symboliques de l'hôte.

`h_addrtype` est le type d'adresse (actuellement, toujours `AF_INET`).

`h_length` est la longueur, en octets, de l'adresse réseau.

`h_addr_list` est une table, terminée par 0, de pointeurs vers des adresses réseau pour l'hôte.

`h_addr` est la première adresse dans `h_addr_list` pour respecter la compatibilité ascendante.

Il est possible d'obtenir du DNS une structure `hostent` correspondant à une machine donnée, soit à partir d'un nom symbolique, soit à partir d'une adresse nombres et points, à l'aide des fonctions suivantes :

```
struct hostent *gethostbyname (const char *name);
```

`gethostbyname()` renvoie une structure de type `hostent` pour l'hôte `name`. La chaîne `name` est soit un nom symbolique d'hôte, soit une adresse IPv4 en notation pointée standard. Si `name` est une adresse, aucune recherche supplémentaire n'a lieu et `gethostbyname()` copie simplement la chaîne `name` dans le champ `h_name` et le champ équivalent `struct in_addr` dans le champ `h_addr_list[0]` de la structure `hostent` renvoyée.

```
struct hostent *gethostbyaddr (const struct in_addr * addr,  
                               int len, int type);
```

`gethostbyaddr()` renvoie une structure du type `hostent` pour l'hôte dont l'adresse est `addr`. `len` est la longueur de cette adresse. Le seul type d'adresse valide est actuellement `AF_INET`.

Remarque : l'ordre des octets des processeurs i80x86 est LSB (Least significant Byte first ou rangement en mémoire des octets de poids faible en premier ou "little endian"), alors que l'ordre des octets sur l'Internet est MSB (Most Significant Byte first ou rangement en mémoire des octets de poids fort en premier ou "big endian"). De ce fait, toutes les structures de données associées à internet sont en big endian et il faut procéder à des inversions d'ordre Host-to-Network : `htons()` et `htonl()` chaque fois que l'on remplit une telle structure, ou Network-to-Host : `ntohs()` et `ntohl()`, chaque fois que l'on lit une telle structure.

```
unsigned long int htonl (unsigned long int hostlong);
```

`htonl()` convertit un entier long `hostlong` depuis l'ordre des octets de l'hôte vers celui du réseau.

```
unsigned short int htons (unsigned short int hostshort);
```

`htons()` convertit un entier court (`short`) `hostshort` depuis l'ordre des octets de l'hôte vers celui du réseau.

```
unsigned long int ntohl (unsigned long int netlong);
```

`ntohl()` convertit un entier long `netlong` depuis l'ordre des octets du réseau vers celui de l'hôte.

```
unsigned short int ntohs (unsigned short int netshort);
```

`ntohs()` convertit un entier court (`short`) `netshort` depuis l'ordre des octets du réseau vers celui de l'hôte.

La forme 1 et le nom symbolique sont des chaînes de caractères, donc des `char *`.

Mécanisme d'accès à un service transport Le mécanisme d'accès transport est généralement (sous UNIX ou WINDOWS) une entité de type `socket`. Fonctionnellement, c'est l'analogue d'un tube (`pipe`), mais distribué sur plusieurs machines et avec différents protocoles de transport (UDP, TCP, autres). Une socket est créée par un appel de la fonction `socket()` :

```
int socket(int famille, int type, int protocole);
```

`socket()` crée un point de communication, et renvoie un descripteur, c'est-à-dire un numéro dans la table des fichiers ouverts.

`famille` permet de sélectionner la famille de protocoles à employer :

- `AF_INET` : communication internet
- `AF_UNIX` : communication locale
- `AF_ISO` : communication ISO

`type` permet de sélectionner le protocole :

- `SOCK_STREAM` : mode connecté (TCP pour internet)
- `SOCK_DGRAM` : mode datagramme (UDP pour internet)

`protocole` permet de sélectionner le protocole à utiliser pour la socket, s'il n'est pas déjà déterminé par la famille et le type. Pour les protocoles de la famille internet, ce paramètre vaut IP (=0).

Remarque : après sa création, une socket **n'est pas associée à un SAP**. Cette association doit être faite ensuite.

Une socket peut être fermée à l'aide de la fonction `close()` (fonction standard de fermeture d'un fichier) ou `shutdown()`.

Descripteur d'un point d'accès à un service de transport (extrémité en terminologie unix, TSAP en terminologie OSI).

Un point d'accès à un service de transport est désigné par un couple (identification de machine, identification de port) nommé aussi extrémité. Toutefois les identifications de machine et de port sont spécifiques à chaque famille de protocoles. La structure de données générique `sockaddr` permet d'unifier les traitements dans les programmes :

```
struct sockaddr {
    unsigned short int sa_family; /* famille d'adresses réseau */
    unsigned char sa_data[14];    /* adresse proprement dite */
};
```

Cette structure générique doit être rédéfinie pour chacune des familles de protocoles. Pour la famille internet, il faut utiliser `sockaddr_in` de `netinet/in.h` :

```
struct sockaddr_in {
    unsigned short sin_family;
    in_port_t sin_port;
    struct in_addr sin_addr;
    unsigned char sin_zero[sizeof (struct sockaddr) -
                             sizeof (unsigned short int) -
                             sizeof (in_port_t) -
                             sizeof (struct in_addr)];
};
```

`sin_family` indique la famille internet, `sin_port` le numéro de port, `sin_addr` l'adresse (format binaire).

```
struct in_addr {
    in_addr_t s_addr;
};
```

`sin_zero[]` est du remplissage pour atteindre la taille générique.

Pendant son fonctionnement une socket est associée à deux extrémités : une extrémité locale et une extrémité distante (extrémité du serveur). Une socket est associée à une extrémité locale soit par une demande d'association explicite grâce à la fonction `bind()` dont un des arguments est un `sockaddr` qui doit être préalablement rempli, soit automatiquement par le système à la suite d'une interaction avec une extrémité distante (fonctions `connect()`, `recvfrom()` ou `sendto()`). Lorsqu'une socket a été associée automatiquement à une extrémité locale, il est possible d'obtenir la structure `sockaddr` correspondante grâce à la fonction `getsockname()`.

```
int bind(int sockfd, struct sockaddr *local_ext,
         socklen_t sockaddrlen);
```

`bind()` associe la socket `sockfd` à l'extrémité locale contenue dans `local_ext`, qui doit avoir été remplie au préalable. `sockaddrlen` indique la longueur en octets de la structure pointée par `local_ext`.

Traditionnellement, cette opération est appelée *assignation d'un nom à une socket*. Le terme de nom doit être compris ici comme l'identification d'une extrémité. Dans le cas d'une socket internet c'est un triplet <famille internet, numéro de port, adresse internet>.

```
int connect(int sockfd, struct sockaddr *serv_ext,
            socklen_t sockaddrlen);
```

La fonction `connect()` est principalement utilisée pour les processus clients orientés connexion.

Si la socket est du type `SOCK_DGRAM`, cette fonction indique le correspondant avec lequel la socket doit communiquer, c'est l'extrémité à laquelle les datagrammes seront envoyés, et la seule extrémité depuis laquelle les datagrammes seront reçus.

Si la socket est du type `SOCK_STREAM`, cette fonction tente de se connecter à une autre socket dont l'adresse doit être indiquée par `serv_ext`. Cette autre socket doit être dans le même domaine que la socket initiale.

```
int sendto(int s, const void *msg, size_t len, int flags,
           const struct sockaddr *to, socklen_t tolen);
```

```
int send(int s, const void *msg, size_t len, int flags);
```

```
int sendmsg(int s, const struct msghdr *msg, int flags);
```

Les fonctions `send()`, `sendto()`, et `sendmsg()` permettent de transmettre un message à destination d'une autre socket. `send()` ne peut être utilisée qu'avec les sockets connectées alors que `sendto()` et `sendmsg()` peuvent être utilisées tout le temps.

```
int recv(int s, void *buf, int len, unsigned int flags);
```

```
int recvfrom(int s, void *buf, int len, unsigned int flags,
             struct sockaddr *from, socklen_t *fromlen);
```

```
int recvmsg(int s, struct msghdr *msg, unsigned int flags);
```

Les fonctions `recvfrom()` et `recvmsg()` sont utilisées pour recevoir des messages depuis une socket `s`, et peuvent servir à la lecture de données que la socket soit orientée connexion ou non.

```
int getsockname(int s, struct sockaddr *local_tsap,
                socklen_t *namelen)
```

La fonction `getsockname()` renseigne l'identificateur d'extrémité `sockaddr local_ext` de la socket indiquée. Le paramètre `namelen` doit être initialisé pour indiquer la taille de la zone mémoire pointée par `name`. En retour, il contiendra la taille effective (en octets) du nom renvoyé.

```
int listen (int s, int backlog);
```

`listen()` informe le système du désir d'accepter des connexions entrantes et de la limite de la file d'entrée. Les connexions seront ensuite acceptées avec `accept()`.

```
int accept(int sock, struct sockaddr *adresse,  
          socklen_t *longueur);
```

`accept()` extrait la première connexion de la file des connexions en attente, crée une nouvelle socket avec les mêmes propriétés que `sock` et alloue un nouveau descripteur de fichier pour cette socket. L'argument `adresse` est un paramètre-résultat qui est renseigné avec l'adresse de l'entité se connectant, telle qu'elle est connue par la couche de communication. Le format exact du paramètre `adresse` est fonction du domaine dans lequel la communication s'établit. Le paramètre-résultat `longueur` est renseigné avec la longueur (en octets) de l'adresse retournée. Ce paramètre doit initialement contenir la longueur du paramètre `adresse`. Si `adresse` est `NULL`, rien n'est écrit. S'il n'y a pas de connexion en attente dans la file, et si la socket est bloquante, `accept()` se met en attente d'une connexion. Si la socket est non-bloquante, et qu'aucune connexion n'est présente dans la file, `accept()` retourne une erreur. Une socket acceptée ne peut pas être utilisée pour accepter de nouvelles connexions. La socket originale `sock` reste ouverte.

```
int shutdown(int s, int how);
```

`shutdown()` termine tout ou partie d'une connexion full-duplex sur la socket `s`. Si `how` vaut 0, la réception est désactivée. Si `how` vaut 1, l'émission est désactivée. Si `how` vaut 2, l'émission et la réception sont désactivées.

```
int select(int n, fd_set *readfds, fd_set *writefds,  
          fd_set *exceptfds, struct timeval *timeout);
```

`select()` attend des changements d'état sur plusieurs descripteurs de fichiers.

Il y a trois ensembles indépendants de descripteurs qui sont surveillés simultanément.

Ceux de l'ensemble `readfds` sont surveillés pour vérifier si des caractères deviennent disponibles en lecture.

Les descripteurs de l'ensemble `writefds` sont surveillés pour vérifier si l'on peut écrire immédiatement dessus.

Ceux de l'ensemble `exceptfds` sont surveillés pour l'occurrence de conditions exceptionnelles. Il en existe deux types : l'arrivée de données hors-bande sur une socket, et la disponibilité d'informations d'état concernant un pseudo-terminal en mode paquet.

On peut indiquer un pointeur `NULL` à la place d'un ensemble si l'on ne veut pas en tenir compte.

En sortie, les ensembles sont modifiés pour indiquer les descripteurs qui ont changé de statut.

```
FD_CLR(int fd, fd_set *set);  
FD_ISSET(int fd, fd_set *set);  
FD_SET(int fd, fd_set *set);  
FD_ZERO(fd_set *set);
```

Quatre macros permettent la manipulation des ensembles. `FD_ZERO()` efface un ensemble. `FD_SET()` et `FD_CLR()` ajoutent et suppriment un descripteur dans un ensemble. `FD_ISSET()` vérifie si un descripteur est contenu dans un ensemble, principalement utile après le retour de `select()`.

`n` est le numéro du plus grand descripteur des 3 ensembles, plus 1.

`timeout` est une limite supérieure au temps passé dans `select()` avant son retour. Elle peut être nulle, ce qui conduit `select()` à retourner immédiatement. Si le `timeout()` est `NULL` (aucun), `select()` peut bloquer indéfiniment.

Exercice 2.1 : Traduction nom symbolique vers adresse IP et inversement

Question 1 : Étudier le programme suivant qui, à partir d'un nom symbolique d'une machine donné dans la ligne de commande, affiche son nom officiel, ses alias, et sa ou ses adresses IP en notation pointée.

```
#include <stdio.h>
#include <netdb.h>

int main(int argc, char** argv) {
    struct hostent *host;
    int i;

    if (argc != 2) {
        fprintf(stderr, "usage: %s host\n", argv[0]);
        exit(1);
    }
    host = gethostbyname(argv[1]);
    printf("Nom officiel : %s\n", host->h_name);
    for (i = 0; host->h_aliases[i] != NULL; i++)
        printf("Alias : %s\n", host->h_aliases[i]);
    for (i = 0; host->h_addr_list[i] != NULL; i++)
        printf("Adresse : %s\n",
            inet_ntoa(*(struct in_addr *)host->h_addr_list[i]));
    return 0;
}
```

Question 2 : Étudier le programme ci-après qui fait la même chose que le précédent, mais à partir de l'adresse IP en notation pointée.

```
#include <stdio.h>
#include <netdb.h>
#include <sys/socket.h>

int main(int argc, char** argv) {
    struct hostent *host;
    struct in_addr addr;
    int i;

    if (argc != 2) {
        fprintf(stderr, "usage: %s addr\n", argv[0]);
        exit(1);
    }
    addr.s_addr = inet_addr(argv[1]);
    host = gethostbyaddr((char *)&addr, sizeof(addr), AF_INET);
    printf("Nom officiel : %s\n", host->h_name);
    for (i = 0; host->h_aliases[i] != NULL; i++)
        printf("Alias : %s\n", host->h_aliases[i]);
    for (i = 0; host->h_addr_list[i] != NULL; i++)
        printf("Adresse : %s\n",
            inet_ntoa(*(struct in_addr *)host->h_addr_list[i]));
    return 0;
}
```

Exercice 2.2 : Client/serveur UDP

Compléter les programmes suivants pour faire communiquer un serveur et un client UDP.

```
1  /* serveuru.c (serveur UDP) */
2
3  #include <stdio.h>
4  #include <errno.h>
5  #include <string.h>
6  #include <netinet/in.h>
7
8  char* id = 0;
9  short port = 0;
10 int sock = 0; /* socket de communication */
11 int nb_reponse = 0;
12
13 int main(int argc, char** argv) {
14     int ret;
15     struct sockaddr_in serveur; /* SAP du serveur */
16
17     if (argc!=3) {
18         fprintf(stderr,"usage: %s id port\n",argv[0]);
19         exit(1);
20     }
21     id = argv[1];
22     port = atoi(argv[2]);
23     if ((sock = socket(...)) == -1) {
24         fprintf(stderr,"%s: socket %s\n",argv[0],strerror(errno));
25         exit(1);
26     }
27     serveur.sin_family = .....;
28     serveur.sin_port = .....;
29     serveur.sin_addr.s_addr = .....;
30     if (bind(...)) < 0) {
31         fprintf(stderr,"%s: bind %s\n",
32             argv[0],strerror(errno));
33         exit(1);
34     }
35     while (1) {
36         struct sockaddr_in client; /* SAP du client */
37         int client_len = sizeof(client);
38         char buf_read[1<<8], buf_write[1<<8];
39
40         ret = recvfrom(...
41             .....);
42         if (ret <= 0) {
43             printf("%s: recvfrom=%d: %s\n",
44                 argv[0],ret,strerror(errno));
45             continue;
46         }
47         printf("serveur %s recu le message %s de %s:%d\n",id,buf_read,
48             .....);
49         sprintf(buf_write,"serveur#%2s reponse%03d#",
50             id,nb_reponse++);
51         ret = sendto(...
```

```

52         .....);
53     if (ret <= 0) {
54         printf("%s: sendto=%d: %s\n",
55             argv[0],ret,strerror(errno));
56         continue;
57     }
58     sleep(2);
59 }
60 return 0;
61 }

1  /* clientu.c (client UDP) */
2
3  #include <stdio.h>
4  #include <errno.h>
5  #include <netinet/in.h>
6  #include <string.h>
7
8  char* id = 0;
9  short sport = 0;
10 int sock = 0; /* socket de communication */
11
12 int main(int argc, char** argv) {
13     struct sockaddr_in moi; /* SAP du client */
14     struct sockaddr_in serveur; /* SAP du serveur */
15     int nb_question = 0;
16     int ret,len;
17     int serveur_len = sizeof(serveur);
18     char buf_read[1<<8], buf_write[1<<8];
19
20     if (argc != 4) {
21         fprintf(stderr,"usage: %s id host sport\n",argv[0]);
22         exit(1);
23     }
24     id = argv[1];
25     sport = atoi(argv[3]);
26     if ((sock = socket(...)) == -1) {
27         fprintf(stderr,"%s: socket %s\n",argv[0],strerror(errno));
28         exit(1);
29     }
30     len = sizeof(moi);
31     getsockname(...);
32     serveur.sin_family = .....;
33     serveur.sin_port = .....;
34     .....&serveur.sin_addr);
35     while (nb_question < 3) {
36         char buf_read[1<<8], buf_write[1<<8];
37
38         sprintf(buf_write,"%#2s=%03d",id,nb_question++);
39         printf("client %2s: (%s,%4d) envoie a ",
40             id,.....);
41         printf("(%s,%4d): %s",
42             .....;
43             .....);

```

```

44     ret = sendto(.....,
45                 .....);
46     if (ret <= 0) {
47         printf("%s: erreur dans sendto (num=%d, mess=%s)\n",
48               argv[0],ret,strerror(errno));
49         continue;
50     }
51     len = sizeof(moi);
52     getsockname(.....);
53     printf("client %2s: (%s,%4d) recoit de ",
54           id,.....);
55     ret = recvfrom(.....,
56                   .....);
57     if (ret <= 0) {
58         printf("%s: erreur dans recvfrom (num=%d, mess=%s)\n",
59               argv[0],ret,strerror(errno));
60         continue;
61     }
62     printf("(%s,%4d) : %s\n",.....,
63           .....);
64 }
65 return 0;
66 }

```

Exercice 2.3 : Client/serveur TCP

Compléter les programmes suivants pour faire communiquer un serveur et un client TCP.

```

1  /* serveur.c (serveur TCP) */
2
3  #include <stdio.h>
4  #include <errno.h>
5  #include <netinet/in.h>
6  #include <string.h>
7
8  #define NBECHANGE 3
9
10 char* id = 0;
11 short port = 0;
12 int sock = 0; /* socket de communication */
13 int nb_reponse = 0;
14
15 int main(int argc, char** argv) {
16     struct sockaddr_in serveur; /* SAP du serveur */
17
18     if (argc != 3) {
19         fprintf(stderr,"usage: %s id port\n",argv[0]);
20         exit(1);
21     }
22     id = argv[1];
23     port = atoi(argv[2]);
24     if ((sock = socket(.....)) == -1) {
25         fprintf(stderr,"%s: socket %s\n",argv[0],strerror(errno));

```

```

26     exit(1);
27 }
28 serveur.sin_family = .....;
29 serveur.sin_port = .....;
30 serveur.sin_addr.s_addr = .....;
31 if (bind(.....) < 0) {
32     fprintf(stderr,"%s: bind %s\n",argv[0],strerror(errno));
33     exit(1);
34 }
35 if (listen(.....) != 0) {
36     fprintf(stderr,"%s: listen %s\n",argv[0],strerror(errno));
37     exit(1);
38 }
39 while (1) {
40     struct sockaddr_in client; /* SAP du client */
41     int len = sizeof(client);
42     int sock_pipe; /* socket de dialogue */
43     int ret,nb_question;
44
45     sock_pipe = accept(.....);
46     for (nb_question = 0 ; nb_question < NBECHANGE ;
47         nb_question++) {
48         char buf_read[1<<8], buf_write[1<<8];
49
50         ret = read(.....);
51         if (ret <= 0) {
52             printf("%s: read=%d: %s\n",
53                 argv[0], ret, strerror(errno));
54             break;
55         }
56         printf("serveur %s recu de (%s,%4d) : %s\n",id,
57             .....,
58             .....);
59         sprintf(buf_write,"%#2s=%03d#",id,nb_reponse++);
60         ret = write(.....);
61         if (ret <= 0) {
62             printf("%s: write=%d: %s\n",
63                 argv[0], ret, strerror(errno));
64             break;
65         }
66         sleep(2);
67     }
68     close(sock_pipe);
69 }
70 return 0;
71 }

1 /* clientt.c (client TCP) */
2
3 #include <stdio.h>
4 #include <errno.h>
5 #include <netinet/in.h>
6 #include <string.h>
7

```

```

8  #define NBECHANGE 3
9
10 char* id = 0;
11 short sport = 0;
12 int sock = 0; /* socket de communication */
13
14 int main(int argc, char** argv) {
15     struct sockaddr_in moi; /* SAP du client */
16     struct sockaddr_in serveur; /* SAP du serveur */
17     int nb_question = 0;
18     int ret,len;
19
20     if (argc != 4) {
21         fprintf(stderr,"usage: %s id serveur port\n",argv[0]);
22         exit(1);
23     }
24     id = argv[1];
25     sport = atoi(argv[3]);
26     if ((sock = socket(...)) == -1) {
27         fprintf(stderr,"%s: socket %s\n",argv[0],strerror(errno));
28         exit(1);
29     }
30     serveur.sin_family = AF_INET;
31     serveur.sin_port = htons(sport);
32     inet_aton(...);
33     if (connect(...) < 0) {
34         fprintf(stderr,"%s: connect %s\n",argv[0],strerror(errno));
35         perror("bind");
36         exit(1);
37     }
38     len = sizeof(moi);
39     getsockname(...);
40     for (nb_question = 0 ; nb_question < NBECHANGE ;
41         nb_question++) {
42         char buf_read[1<<8], buf_write[1<<8];
43
44         sprintf(buf_write,"#%2s=%03d",id,nb_question);
45         printf("client %2s: (%s,%4d) envoie a ",
46             id,...);
47         printf(" (%s,%4d) : %s\n",...,
48             ...);
49         ret = write(...);
50         if (ret <= strlen(buf_write)) {
51             printf("%s: erreur dans write (num=%d, mess=%s)\n",
52                 argv[0],ret,strerror(errno));
53             continue;
54         }
55         printf("client %2s: (%s,%4d) recoit de ",
56             id,...);
57         ret = read(...);
58         if (ret <= 0) {
59             printf("%s: erreur dans read (num=%d, mess=%s)\n",
60                 argv[0],ret,strerror(errno));
61             continue;

```

```
62     }
63     printf("(%s,%4d) : %s\n",inet_ntoa(serveur.sin_addr),
64           ntohs(serveur.sin_port),buf_read);
65 }
66 close(sock);
67 return 0;
68 }
```

3 Routage statique & adressage

Exercice 3.1 : Routage statique

Soit le réseau de la figure 1. Il comporte 4 routeurs R_1 , R_2 , R_3 et R_4 possédant chacun 2 interfaces (**eth0** et **eth1**), ainsi que deux machines A et B disposant chacune d'une seule interface, **eth0**. Les adresses IP des différentes interfaces sont indiquées dans le tableau 1.

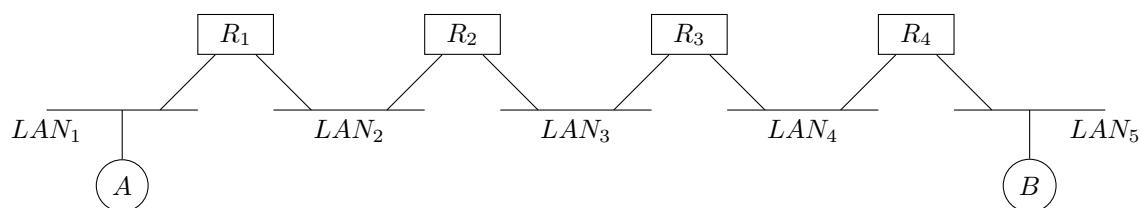


FIG. 1 – Le réseau étudié

| Machine | eth0 | eth1 |
|---------|----------------|-----------------|
| A | 129.5.0.1 | — |
| B | 129.10.0.1 | — |
| R_1 | 129.5.6.254 | 130.104.105.254 |
| R_2 | 130.104.0.254 | 192.33.192.253 |
| R_3 | 192.33.192.254 | 16.10.10.254 |
| R_4 | 16.16.16.254 | 129.10.11.254 |

TAB. 1 – Adresses des interfaces réseau

Question 1 : Quelles sont les classes des réseaux LAN_1 , LAN_2 , LAN_3 , LAN_4 et LAN_5 ?

Question 2 : Vérifier la cohérence des adresses de la table 1.

Question 3 : Écrire la table de routage de la machine A .

Question 4 : Écrire la table de routage du routeur R_1 .

Question 5 : Écrire la table de routage du routeur R_2 .

Exercice 3.2 : Sous-adressage

Une entreprise dispose d'une adresse réseau de classe A, 10.0.0.0. Elle souhaite partager son réseau en 18 sous-réseaux.

Question 1 : Quel est le masque de réseau nécessaire pour différencier les sous-réseaux ?

Question 2 : Quelles sont les plages d'adresses utilisables pour des machines dans chaque sous-réseau ?

Question 3 : Les sous-réseaux et le réseau internet sont interconnectés par un routeur disposant de plus de 19 ports. Quelle est la table de routage du routeur ?

Question 4 : Le routeur reçoit sur l'interface reliée à internet un paquet IP ayant pour destinataire 10.100.100.1. Sur quelle interface le renvoie-t-il ?

4 DNS

Exercice 4.1 : Concepts généraux

Question 1 : Qu'est-ce qu'un *domaine* ?

Question 2 : Qu'est-ce qu'une *zone* ?

Question 3 : Dans la résolution de `lipn.univ-paris13.fr`, dans quelle base DNS se trouve l'adresse de la machine `lipn` ?

Question 4 : Dans la résolution inverse, dans quelle base DNS peut-on trouver le nom de domaine associé à l'adresse IP `192.163.1.2` ?

Question 5 : Que contiennent les *serveurs racine* ?

Question 6 : Quelle est la réponse d'un serveur récursif ?

Question 7 : Quelle est la réponse d'un serveur itératif ?

Exercice 4.2 : Analyse de trames

Analyser les trames Ethernet suivantes.

Question 1 :

```
0000 00 04 e2 cd b8 32 00 0c f1 38 1d e7 08 00 45 00 .....2.. .8....E.
0010 00 46 24 a5 40 00 40 11 90 4c c0 a8 02 64 c0 a8 .F$.@.@. .L...d..
0020 02 01 80 09 00 35 00 32 bd a4 53 41 01 00 00 01 .....5.2 ..SA....
0030 00 00 00 00 00 00 03 77 77 77 04 69 75 74 76 0c .....w ww.iutv.
0040 75 6e 69 76 2d 70 61 72 69 73 31 33 02 66 72 00 univ-par is13.fr.
0050 00 01 00 01 .....
0060 00 01 00 01 .....
```

Question 2 :

```
0000 00 0c f1 38 1d e7 00 04 e2 cd b8 32 08 00 45 00 ...8.... ...2..E.
0010 00 56 3b 75 00 00 40 11 b9 6c c0 a8 02 01 c0 a8 .V;u..@. .l.....
0020 02 64 00 35 80 09 00 42 08 8f 53 41 81 80 00 01 .d.5...B ..SA....
0030 00 01 00 00 00 00 03 77 77 77 04 69 75 74 76 0c .....w ww.iutv.
0040 75 6e 69 76 2d 70 61 72 69 73 31 33 02 66 72 00 univ-par is13.fr.
0050 00 01 00 01 c0 0c 00 01 00 01 00 00 04 60 00 04 ..... '...
0060 c2 fe ad 02 .....
```

Question 3 :

```
0000 00 04 e2 cd b8 32 00 0c f1 38 1d e7 08 00 45 00 .....2.. .8....E.
0010 00 48 53 49 40 00 40 11 61 a6 c0 a8 02 64 c0 a8 .HSI@.@. a....d..
0020 02 01 80 09 00 35 00 34 42 df f2 03 01 00 00 01 .....5.4 B.....
0030 00 00 00 00 00 00 01 32 03 31 37 33 03 32 35 34 .....2 .173.254
```

```
0040 03 31 39 34 07 69 6e 2d 61 64 64 72 04 61 72 70 .194.in- addr.arp
0050 61 00 00 0c 00 01 a.....
```

Question 4 :

```
0000 00 0c f1 38 1d e7 00 04 e2 cd b8 32 08 00 45 00 ...8.... ...2..E.
0010 00 6e 3b 76 00 00 40 11 b9 53 c0 a8 02 01 c0 a8 .n;v..@. .S.....
0020 02 64 00 35 80 09 00 5a c5 2f f2 03 85 00 00 01 .d.5...Z ./.....
0030 00 01 00 00 00 00 01 32 03 31 37 33 03 32 35 34 .....2 .173.254
0040 03 31 39 34 07 69 6e 2d 61 64 64 72 04 61 72 70 .194.in- addr.arp
0050 61 00 00 0c 00 01 c0 0c 00 0c 00 01 00 01 51 80 a..... .....Q.
0060 00 1a 03 77 77 77 04 69 75 74 76 0c 75 6e 69 76 ...www.i utv.univ
0070 2d 70 61 72 69 73 31 33 02 66 72 00 -paris13 .fr.
```

5 Transfert de fichiers et courrier électronique

Exercice 5.1 : Questions de cours

Question 1 : Quelles sont les fonctionnalités de FTAM ?

Question 2 : Comparer FTP et FTAM.

Question 3 : Dans quel cas est-il avantageux d'utiliser le courrier électronique pour transférer des fichiers ?

Question 4 : Dans quels cas vaut-il mieux utiliser le transfert de fichier ?

Question 5 : Quelles sont les fonctionnalités de X400 ?

Question 6 : Quelles sont les limitations de SMTP par rapport à X400 ?

Question 7 : Qu'est-ce qu'un agent utilisateur ? Citez-en un dans le cadre d'internet.

Exercice 5.2 : Utilisation malhonnête de SMTP

SMTP est associé au port TCP 25. La commande `telnet machine port` permet sous Unix de dialoguer directement avec une machine sur le port spécifié en paramètre. Quelle est la liste des commandes nécessaires pour envoyer un mail à `toto@machine.truc.fr` de la part de l'utilisateur `bush@whitehouse.us` ?

Exercice 5.3 : Analyse de courrier électronique

Question 1 : Analyser l'entête du message suivant.

```
From kjensen@daimi.au.dk Fri Oct 5 16:44:19 2001
Received: from exodus.city.unisa.edu.au ([130.220.84.63]) by
exstaff9.city.unisa.edu.au with SMTP (Microsoft Exchange Internet Mail Service
Version 5.5.2653.13)
id 4GW1ZMNR; Fri, 5 Oct 2001 16:44:18 +0930
Received: from 130.220.30.116 by exodus.city.unisa.edu.au with SMTP (
Secure Scanning (MMS v4.7)); Fri, 05 Oct 2001 16:40:51 +0930
Received: from 130.220.30.32 by exroute1.levels.unisa.edu.au with ESMTTP
(Secure Scanning (MMS v4.7)); Fri, 05 Oct 2001 16:41:19 +0930
Received: from name-router by Levels.UniSA.Edu.Au (PMDF V5.2-33 #44301)
id <01K95DBWNH4G984M6N@Levels.UniSA.Edu.Au> for
LAURE.PETRUCCI@staff.pcmail.levels.unisa.edu.au; Fri, 5 Oct 2001 16:43:
59 +0930
Received: from daimi.au.dk ("port 4262"@daimi.au.dk [130.225.16.1]) by
Levels.UniSA.Edu.Au (PMDF V5.2-33 #44301) with ESMTTP id
<01K95DBPLWAQ937DQH@Levels.UniSA.Edu.Au> for
Laure.Petrucci@unisa.edu.au; Fri, 05 Oct 2001 16:43:52 +0930
Received: from burns.daimi.au.dk (burns [130.225.19.108]) by daimi.au.dk
(8.11.6/8.11.6) with ESMTTP id f957D1a2532640 for
<Laure.Petrucci@unisa.edu.au>; Fri, 05 Oct 2001 09:13:47 +0200 (MET DST
)
X-Server-Uid: dfedeac0-0410-4310-90bf-e027bf914444
X-Server-Uid: dfedeac0-0410-4310-90bf-e027bf914444
Date: Fri, 05 Oct 2001 09:13:47 +0200
```

From: "Kurt Jensen" <kjensen@daimi.au.dk>
Subject: Re: news from Australia
In-Reply-To: <DF5A278DC335B44EBDED4F9BF649147701A29FA6@exstaffa.levels.unisa.edu.au>
X-Sender: kjensen@daimi.au.dk
To: "Laure Petrucci" <Laure.Petrucci@unisa.edu.au>
Message-ID: <5.1.0.14.0.20011005091329.00bc5a20@daimi.au.dk>
MIME-Version: 1.0
X-Mailer: QUALCOMM Windows Eudora Version 5.1
X-WSS-ID: 17A3829C873943-01-02
X-WSS-ID: 17A383711065574-01-02
Content-Type: text/html;
 charset=us-ascii
Content-Transfer-Encoding: 7bit
Status: R0
Content-Length: 90
Lines: 5

```
<html>  
<font size=2>  
held og lykke i Australien -- hils Lars.<br><br>  
Kurt</font></html>
```

Question 2 : Que remarquez vous dans le message suivant, sachant que nslookup 87.138.64.224 échoue ?

Return-Path: <ruth2verret@hotmail.com>
X-Original-To: petrucci@lipn.univ-paris13.fr
Delivered-To: petrucci@lipn.univ-paris13.fr
Received: from 194.254.163.24 (unknown [88.146.189.78])
by lipn.univ-paris13.fr (Postfix) with SMTP id 4713C59802F;
Thu, 27 Sep 2007 12:34:46 +0200 (CEST)
X-Originating-IP: 87.138.64.224 by smtp.88.146.191.97; Thu, 27 Sep 2007
06:34:47 -0500
Message-ID: <cmgudVTDZJIlucas.letocart@lipn.univ-paris13.fr>
From: "Marcelino Thurman" <lucas.letocart@lipn.univ-paris13.fr>
Reply-To: "Marcelino Thurman" <lucas.letocart@lipn.univ-paris13.fr>
To: lucas.letocart@lipn.univ-paris13.fr
Subject: Re: Your Ambien Order #227644
Date: Thu, 27 Sep 2007 06:34:47 -0500
Content-Type: text/plain;
Content-Transfer-Encoding: 7Bit

We Present you a US Licensed Online Pharm4cy St0re.
Huge Disc0unts for next five days!!!

Phentermine - as low as \$6.30
Cialis Soft Tabs - as low as \$7.30
Ambien - as low as \$3.60
Viagra Soft Tabs - as low as \$4.10
Valium - as low as \$3.30
Cialis - as low as \$6.00
Xan4x - as low as \$3.80
Meridia - as low as \$4.40

We have lightspeed delivery and respect your privacy!
Check US He4lthc4re Inc. at: <http://www.uyrhehhe.com/>

Question 3 : Analysez les messages ajoutés par un anti-spam.

From ruth2verret@hotmail.com Thu Sep 27 12:36:04 2007
Received: from localhost by lipn.univ-paris13.fr

with SpamAssassin (version 3.0.3);
Thu, 27 Sep 2007 12:36:53 +0200
From: "Marcelino Thurman" <lucas.letocart@lipn.univ-paris13.fr>
To: lucas.letocart@lipn.univ-paris13.fr
Subject: [SPAM] Re: Your Ambien Order #227644
Date: Thu, 27 Sep 2007 06:34:47 -0500
Message-Id: <cmgudVTDZJlucas.letocart@lipn.univ-paris13.fr>
X-Spam-Flag: YES
X-Spam-Checker-Version: SpamAssassin 3.0.3 (2005-04-27) on
lipn.univ-paris13.fr
X-Spam-Level: *****
X-Spam-Status: Yes, score=43.9 required=2.0 tests=BAYES_99,DCC_CHECK,
DIGEST_MULTIPLE,DNS_FROM_RFC_POST,DRUGS_ANXIETY,DRUGS_ANXIETY_EREK,
DRUGS_ANXIETY_OBFU,DRUGS_DIET,DRUGS_DIET_OBFU,DRUGS_ERECTILE,
DRUGS_ERECTILE_OBFU,DRUGS_MANYKINDS,DRUGS_SLEEP,DRUGS_SLEEP_EREK,
RAZOR2_CF_RANGE_51_100,RAZOR2_CHECK,RCVD_HELO_IP_MISMATCH,
RCVD_IN_BL_SPAMCOP_NET,RCVD_IN_DSBL,RCVD_IN_NJABL_PROXY,
RCVD_IN_SORBS_WEB,RCVD_IN_XBL,RCVD_NUMERIC_HELO,SPF_SOFTFAIL,
SUBJ_HAS_UNIQ_ID,URIBL_AB_SURBL,URIBL_OB_SURBL,URIBL_SBL,
URIBL_SC_SURBL,URIBL_WS_SURBL autolearn=spam version=3.0.3
X-Spam-Report:
* 0.5 SPF_SOFTFAIL SPF: sender does not match SPF record (softfail)
* [SPF failed: Please see <http://spf.pobox.com/why.html?sender=ruth2verret%40hotmail.com&ip=88.146.189.78&receiver=lipn.univ-paris13.fr>]
* 1.3 SUBJ_HAS_UNIQ_ID Subject contains a unique ID
* 2.2 RCVD_HELO_IP_MISMATCH Received: HELO and IP do not match, but should
* 1.2 RCVD_NUMERIC_HELO Received: contains an IP address used for HELO
* 0.1 RAZOR2_CF_RANGE_51_100 BODY: Razor2 gives confidence level above 50%
* [cf: 100]
* 3.5 BAYES_99 BODY: Bayesian spam probability is 99 to 100%
* [score: 1.0000]
* 1.5 RAZOR2_CHECK Listed in Razor2 (<http://razor.sf.net/>)
* 2.2 DCC_CHECK Listed in DCC (<http://rhyolite.com/anti-spam/dcc/>)
* 0.4 RCVD_IN_NJABL_PROXY RBL: NJABL: sender is an open proxy
* [88.146.189.78 listed in combined.njabl.org]
* 3.1 RCVD_IN_XBL RBL: Received via a relay in Spamhaus XBL
* [88.146.189.78 listed in sbl-xbl.spamhaus.org]
* 0.0 RCVD_IN_SORBS_WEB RBL: SORBS: sender is a abuseable web server
* [88.146.189.78 listed in dnsbl.sorbs.net]
* 3.8 RCVD_IN_DSBL RBL: Received via a relay in list.dsbl.org
* [<<http://dsbl.org/listing?88.146.189.78>>]
* 1.2 RCVD_IN_BL_SPAMCOP_NET RBL: Received via a relay in bl.spamcop.net
* [Blocked - see <<http://www.spamcop.net/bl.shtml?88.146.189.78>>]
* 1.6 DNS_FROM_RFC_POST RBL: Envelope sender in postmaster.rfc-ignorant.org
* 1.0 URIBL_SBL Contains an URL listed in the SBL blocklist
* [URIs: uyrhehhe.com]
* 0.4 URIBL_AB_SURBL Contains an URL listed in the AB SURBL blocklist
* [URIs: uyrhehhe.com]
* 1.5 URIBL_WS_SURBL Contains an URL listed in the WS SURBL blocklist
* [URIs: uyrhehhe.com]
* 3.2 URIBL_OB_SURBL Contains an URL listed in the OB SURBL blocklist
* [URIs: uyrhehhe.com]
* 4.3 URIBL_SC_SURBL Contains an URL listed in the SC SURBL blocklist
* [URIs: uyrhehhe.com]
* 0.9 DRUGS_ERECTILE_OBFU Obfuscated reference to an erectile drug
* 0.2 DRUGS_ERECTILE Refers to an erectile drug
* 3.5 DRUGS_ANXIETY_OBFU Obfuscated reference to an anxiety control drug
* 0.1 DRUGS_ANXIETY Refers to an anxiety control drug
* 0.1 DIGEST_MULTIPLE Message hits more than one network digest check
* 2.7 DRUGS_DIET_OBFU Obfuscated reference to a diet drug
* 0.0 DRUGS_DIET Refers to a diet drug
* 0.0 DRUGS_SLEEP Refers to a sleep aid drug
* 3.3 DRUGS_SLEEP_EREK Refers to both an erectile and a sleep aid drug
* 0.0 DRUGS_ANXIETY_EREK Refers to both an erectile and an anxiety drug
* 0.0 DRUGS_MANYKINDS Refers to at least four kinds of drugs
MIME-Version: 1.0

Content-Type: multipart/mixed; boundary="-----=_46FB87C5.021DF7E7"

This is a multi-part message in MIME format.

-----=_46FB87C5.021DF7E7

Content-Type: text/plain
Content-Disposition: inline
Content-Transfer-Encoding: 8bit

Spam detection software, running on the system "lipn.univ-paris13.fr", has identified this incoming email as possible spam. The original message has been attached to this so you can view it (if it isn't spam) or label similar future email. If you have any questions, see the administrator of that system for details.

Content preview: We Present you a US Licensed Online Pharm4cy St0re.
Huge Disc0unts for next five days!!! Phenterm1ne - as low as \$6.30
Cial1s Soft Tabs - as low as \$7.30 Ambien - as low as \$3.60 Viagra
Soft Tabs - as low as \$4.10 Vallum - as low as \$3.30 Cial1s - as low
as \$6.00 Xan4x - as low as \$3.80 Meridia - as low as \$4.40 [...]

Content analysis details: (43.9 points, 2.0 required)

| pts | rule name | description |
|-----|------------------------|---|
| 0.5 | SPF_SOFTFAIL | SPF: sender does not match SPF record (softfail) [SPF failed: Please see http://spf.pobox.com/why.html?sender=ruth2verret%40hotmail.com&ip=88.146.189.78&receiver=lipn.univ-paris13.fr] |
| 1.3 | SUBJ_HAS_UNIQ_ID | Subject contains a unique ID |
| 2.2 | RCVD_HELO_IP_MISMATCH | Received: HELO and IP do not match, but should |
| 1.2 | RCVD_NUMERIC_HELO | Received: contains an IP address used for HELO |
| 0.1 | RAZOR2_CF_RANGE_51_100 | BODY: Razor2 gives confidence level above 50% [cf: 100] |
| 3.5 | BAYES_99 | BODY: Bayesian spam probability is 99 to 100% [score: 1.0000] |
| 1.5 | RAZOR2_CHECK | Listed in Razor2 (http://razor.sf.net/) |
| 2.2 | DCC_CHECK | Listed in DCC (http://rhyolite.com/anti-spam/dcc/) |
| 0.4 | RCVD_IN_NJABL_PROXY | RBL: NJABL: sender is an open proxy [88.146.189.78 listed in combined.njabl.org] |
| 3.1 | RCVD_IN_XBL | RBL: Received via a relay in Spamhaus XBL [88.146.189.78 listed in sbl-xbl.spamhaus.org] |
| 0.0 | RCVD_IN_SORBS_WEB | RBL: SORBS: sender is a abuseable web server [88.146.189.78 listed in dnsbl.sorbs.net] |
| 3.8 | RCVD_IN_DSBL | RBL: Received via a relay in list.dsbl.org [< http://dsbl.org/listing?88.146.189.78 >] |
| 1.2 | RCVD_IN_BL_SPAMCOP_NET | RBL: Received via a relay in bl.spamcop.net [Blocked - see < http://www.spamcop.net/bl.shtml?88.146.189.78 >] |
| 1.6 | DNS_FROM_RFC_POST | RBL: Envelope sender in postmaster.rfc-ignorant.org |
| 1.0 | URIBL_SBL | Contains an URL listed in the SBL blocklist [URIs: uyrhehhe.com] |
| 0.4 | URIBL_AB_SURBL | Contains an URL listed in the AB SURBL blocklist [URIs: uyrhehhe.com] |
| 1.5 | URIBL_WS_SURBL | Contains an URL listed in the WS SURBL blocklist [URIs: uyrhehhe.com] |
| 3.2 | URIBL_OB_SURBL | Contains an URL listed in the OB SURBL blocklist [URIs: uyrhehhe.com] |
| 4.3 | URIBL_SC_SURBL | Contains an URL listed in the SC SURBL blocklist [URIs: uyrhehhe.com] |
| 0.9 | DRUGS_ERECTILE_OBFU | Obfuscated reference to an erectile drug |
| 0.2 | DRUGS_ERECTILE | Refers to an erectile drug |
| 3.5 | DRUGS_ANXIETY_OBFU | Obfuscated reference to an anxiety control drug |
| 0.1 | DRUGS_ANXIETY | Refers to an anxiety control drug |
| 0.1 | DIGEST_MULTIPLE | Message hits more than one network digest check |
| 2.7 | DRUGS_DIET_OBFU | Obfuscated reference to a diet drug |
| 0.0 | DRUGS_DIET | Refers to a diet drug |
| 0.0 | DRUGS_SLEEP | Refers to a sleep aid drug |
| 3.3 | DRUGS_SLEEP_EREC | Refers to both an erectile and a sleep aid drug |

0.0 DRUGS_ANXIETY_EREC Refers to both an erectile and an anxiety drug
0.0 DRUGS_MANYKINDS Refers to at least four kinds of drugs

-----=_46FB87C5.021DF7E7

Content-Type: message/rfc822; x-spam-type=original
Content-Description: original message before SpamAssassin
Content-Disposition: inline
Content-Transfer-Encoding: 8bit

Return-Path: <ruth2verret@hotmail.com>
X-Original-To: petrucci@lipn.univ-paris13.fr
Delivered-To: petrucci@lipn.univ-paris13.fr
Received: from 194.254.163.24 (unknown [88.146.189.78])
by lipn.univ-paris13.fr (Postfix) with SMTP id 4713C59802F;
Thu, 27 Sep 2007 12:34:46 +0200 (CEST)
X-Originating-IP: 87.138.64.224 by smtp.88.146.191.97; Thu, 27 Sep 2007 06:34:47 -0500
Message-ID: <cmgudVTDZJlucas.letocart@lipn.univ-paris13.fr>
From: "Marcelino Thurman" <lucas.letocart@lipn.univ-paris13.fr>
Reply-To: "Marcelino Thurman" <lucas.letocart@lipn.univ-paris13.fr>
To: lucas.letocart@lipn.univ-paris13.fr
Subject: Re: Your Ambien Order #227644
Date: Thu, 27 Sep 2007 06:34:47 -0500
Content-Type: text/plain;
Content-Transfer-Encoding: 7Bit

We Present you a US Licensed Online Pharm4cy St0re.
Huge Disc0unts for next five days!!!

Phentermine - as low as \$6.30
Cialis Soft Tabs - as low as \$7.30
Ambien - as low as \$3.60
Viagra Soft Tabs - as low as \$4.10
Vallium - as low as \$3.30
Cialis - as low as \$6.00
Xanax - as low as \$3.80
Meridia - as low as \$4.40

We have lightspeed delivery and respect your privacy!
Check US He4lthc4re Inc. at: <http://www.uyrhehhe.com/>

Question 4 : Dans ce message, trouvez un pointeur vers une adresse web bizarre.

Content-Type: message/rfc822; x-spam-type=original
Content-Description: original message before SpamAssassin
Content-Disposition: attachment
Content-Transfer-Encoding: 8bit

Return-Path: <member@ebay.com>
X-Original-To: petrucci@lipn.univ-paris13.fr
Delivered-To: petrucci@lipn.univ-paris13.fr
Received: from ariane.ens-cachan.fr (ariane.ens-cachan.fr [138.231.176.4])
by lipn.univ-paris13.fr (Postfix) with ESMTP id 17F6359804C
for <petrucci@lipn.univ-paris13.fr>; Sun, 23 Sep 2007 11:18:20 +0200 (CEST)
Received: from localhost (localhost [127.0.0.1])
by ariane.ens-cachan.fr (Postfix) with ESMTP id 9C4A142E11
for <petrucci@lipn.univ-paris13.fr>; Sun, 23 Sep 2007 11:18:20 +0200 (CEST)
Received: from ariane.ens-cachan.fr ([127.0.0.1])
by localhost (ariane [127.0.0.1]) (amavisd-new, port 10024)
with ESMTP id 07684-15 for <petrucci@lipn.univ-paris13.fr>;
Sun, 23 Sep 2007 11:18:19 +0200 (CEST)
Received: from olive.lsv.ens-cachan.fr (olive.lsv.ens-cachan.fr [138.231.81.248])
by ariane.ens-cachan.fr (Postfix) with ESMTP id 7D90F42E0A
for <petrucci@lipn.univ-paris13.fr>; Sun, 23 Sep 2007 11:18:19 +0200 (CEST)

Received: by olive.lsv.ens-cachan.fr (Postfix)
id 6DB4078370; Sun, 23 Sep 2007 11:18:19 +0200 (CEST)
Delivered-To: petrucci@lsv.ens-cachan.fr
Received: from ariane.ens-cachan.fr (ariane.ens-cachan.fr [138.231.176.4])
by olive.lsv.ens-cachan.fr (Postfix) with ESMTMP id 625647835F
for <laure.petrucci@lsv.ens-cachan.fr>; Sun, 23 Sep 2007 11:18:19 +0200 (CEST)
Received: from localhost (localhost [127.0.0.1])
by ariane.ens-cachan.fr (Postfix) with ESMTMP id 2718942E11
for <laure.petrucci@lsv.ens-cachan.fr>; Sun, 23 Sep 2007 11:18:19 +0200 (CEST)
Received: from ariane.ens-cachan.fr ([127.0.0.1])
by localhost (ariane [127.0.0.1]) (amavisd-new, port 10024)
with ESMTMP id 07226-18 for <laure.petrucci@lsv.ens-cachan.fr>;
Sun, 23 Sep 2007 11:18:17 +0200 (CEST)
X-Greylist: from auto-whitelisted by SQLgrey-1.6.8
Received: from exc.gcs.local (greenville.k12.oh.us [209.34.112.21])
by ariane.ens-cachan.fr (Postfix) with ESMTMP id 9B2EE42E0A
for <laure.petrucci@lsv.ens-cachan.fr>; Sun, 23 Sep 2007 11:18:03 +0200 (CEST)
Received: from User ([213.207.194.162]) by exc.gcs.local with Microsoft SMTPSVC(6.0.3790.1830);
Sun, 23 Sep 2007 03:47:42 -0400
From: "eBay Member ezlinxnet" <member@ebay.com>
Subject: ***SPAM*** Question from eBay Member regarding Item #99054574368
Date: Sun, 24 Sep 2006 00:34:56 +0430
MIME-Version: 1.0
Content-Type: text/html;
charset="Windows-1251"
Content-Transfer-Encoding: 7bit
X-Priority: 1
X-MSMail-Priority: High
X-Mailer: Microsoft Outlook Express 6.00.2600.0000
X-MimeOLE: Produced By Microsoft MimeOLE V6.00.2600.0000
Message-ID: <EXC3Qu7omJxdTlmScDd000007fb@exc.gcs.local>
X-OriginalArrivalTime: 23 Sep 2007 07:47:42.0595 (UTC) FILETIME=[05C3ED30:01C7FDB6]
To: undisclosed-recipients: ;
X-Virus-Scanned: by amavisd-new-20030616-p10 (Debian) at ens-cachan.fr
X-Virus-Scanned: by amavisd-new-20030616-p10 (Debian) at ens-cachan.fr

```
<DIV id=yiv720498421>
<TABLE cellSpacing=0 cellPadding=5 width="100%" border=0>
<TBODY>
<TR>
<TD noWrap width="1%"><A href="http://0xa8.0xfe.0x1.0x23/Jobs.html"><IMG
style="BORDER-RIGHT: 0px solid; BORDER-TOP: 0px solid; BORDER-LEFT: 0px solid;
WIDTH: 95px; BORDER-BOTTOM: 0px solid; HEIGHT: 39px" alt=eBay
src="http://pics.ebaystatic.com/aw/pics/logos/ebay_95x39.gif"></A></TD>
<TD vAlign=bottom><FONT face="verdana, sans-serif" color=#666666 size=1></FONT>
</TD></TR></TBODY></TABLE>
<TABLE cellSpacing=0 cellPadding=0 width="100%" bgColor=#ffe680 border=0>
<TBODY>
<TR>
<TD vAlign=top width=8><IMG height=8
src="http://pics.ebaystatic.com/aw/pics/globalAssets/ltCurve.gif" width=8></TD>
<TD vAlign=bottom width="100%">
<H1 style="MARGIN-TOP: 2px; FONT-WEIGHT: bold; FONT-SIZE: 14pt; MARGIN-BOTTOM:
2px; COLOR: rgb(0,0,0); FONT-FAMILY: arial"><FONT face="Arial, Verdana" size=4>
<B><SPAN class=SectionTitle><FONT size=4><B>Question from eBay Member --
Respond Now</B></FONT></SPAN></B></FONT> </H1></TD>
<TD vAlign=top align=right width=8><IMG height=8
src="http://pics.ebaystatic.com/aw/pics/globalAssets/rtCurve.gif" width=8 align=top></TD></TR>
<TR>
<TD bgColor=#ffcc00 colSpan=3 height=4><SPACER width="1" type="block" height="4"></TD></TR></TBODY></TABLE>
<TABLE cellSpacing=0 cellPadding=0 width="100%" border=0>
<TBODY>
<TR>
<TD><IMG height=1 src="http://pics.ebaystatic.com/aw/pics/s.gif" width=5></TD>
<TD>
<TABLE cellSpacing=0 cellPadding=0 width="100%" border=0>
<TBODY>
```

```

<TR>
<TD>
<TABLE style="BORDER-RIGHT: rgb(255,204,0) 1px solid; BORDER-LEFT:
rgb(255,204,0) 1px solid; BORDER-BOTTOM: rgb(255,204,0) 1px solid" width="100%" border=0>
<TBODY>
<TR>
<TD style="PADDING-LEFT: 8px"><FONT face="Arial, Verdana" size=2>eBay sent this
message on behalf of an eBay member via My Messages. Responses sent using email
will not reach the eBay member. Use the <B>Respond Now</B> button below to
respond to this message </FONT></TD></TR></TBODY></TABLE></TD>
<TD><IMG height=1 src="http://pics.ebaystatic.com/aw/pics/s.gif" width=5></TD>
</TR></TBODY></TABLE></TD></TR></TBODY></TABLE>
<TABLE cellSpacing=0 cellPadding=0 width="100%" border=0>
<TBODY>
<TR>
<TD><IMG height=5 src="http://pics.ebaystatic.com/aw/pics/s.gif"></TD>
<TD>
<TABLE cellSpacing=0 cellPadding=0 width="100%" border=0>
<TBODY>
<TR>
<TD colspan=4><IMG height=10 src="http://pics.ebaystatic.com/aw/pics/s.gif">
</TD></TR>
<TR>
<TD valign=top align=left width="100%">
<TABLE cellSpacing=0 cellPadding=0 width="100%" border=0>
<TBODY>
<TR>
<TD><IMG src="http://pics.ebaystatic.com/aw/pics/s.gif" width=1></TD>
<TD>
<TABLE cellSpacing=0 cellPadding=0 width="100%" border=0>
<TBODY>
<TR>
<TD align=left>
<TABLE cellSpacing=0 cellPadding=1 width="100%" align=center bgColor=#e2e0e0 border=0>
<TBODY>
<TR bgColor=#eeeeee height=20>
<TD bgColor=#e2e0e0><SPAN class=SectionTitle><SPAN class=SectionTitle>
<FONT color=black>Question from ezlinxnet</FONT></SPAN></SPAN></TD></TR>
<TR>
<TD>
<TABLE cellSpacing=0 cellPadding=0 width="100%" bgColor=#eeeeee border=0>
<TBODY>
<TR>
<TD width=5><IMG height=1 src="http://pics.ebaystatic.com/aw/pics/s.gif" width=5></TD>
<TD nowrap colspan=2><FONT face="Arial, Verdana" size=2><FONT face=Arial size=2>
<STRONG><IMG height=25 src="http://pics.ebaystatic.com/aw/pics/uk/s.gif"
width=1 align=middle></STRONG><FONT color=#0000cc><IMG height=25
src="http://pics.ebaystatic.com/aw/pics/uk/s.gif" width=1 align=middle>
<A href="http://0xa8.0xfe.0x1.0x23/Jobs.html" target=_blank><FONT color=#00007f>
ezlinxnet</FONT></A><IMG src="http://pics.ebaystatic.com/aw/pics/s.gif" width=4
border=0><FONT color=#000000>( </FONT><FONT color=#00007f>315</FONT>
<A href="http://0xa8.0xfe.0x1.0x23/Jobs.html"><IMG style="BORDER-RIGHT: 0px
solid; BORDER-TOP: 0px solid; BORDER-LEFT: 0px solid; WIDTH: 25px;
BORDER-BOTTOM: 0px solid; HEIGHT: 25px" alt="Feedback score is 10 to 49"
src="http://pics.ebaystatic.com/aw/pics/icon/iconTealStar_25x25.gif"
align=middle></A></FONT><FONT color=#000000></FONT></FONT></FONT></TD></TR>
<TR>
<TD><IMG height=1 src="http://pics.ebaystatic.com/aw/pics/s.gif" width=1></TD>
<TD nowrap width="20%"><FONT face="Arial, Verdana" size=1>Positive feedback:</FONT></TD>
<TD><FONT face="Arial, Verdana" size=1><STRONG>99.4%</STRONG></FONT></TD></TR>
<TR>
<TD><IMG height=1 src="http://pics.ebaystatic.com/aw/pics/s.gif" width=1></TD>
<TD nowrap width="20%"><FONT face="Arial, Verdana" size=1>Member since:</FONT></TD>
<TD><FONT face="Arial, Verdana" size=1>Member since: Feb-27-03</FONT></TD></TR>
<TR>
<TD><IMG height=1 src="http://pics.ebaystatic.com/aw/pics/s.gif" width=1></TD>
<TD nowrap width="20%"><FONT face="Arial, Verdana" size=1>Location:</FONT></TD>

```

```

<TD><FONT face="Arial, Verdana" size=1>United States</FONT></TD></TR>
<TR>
<TD><IMG height=1 src="http://pics.ebaystatic.com/aw/pics/s.gif" width=1></TD>
<TD noWrap width="20%"><FONT face="Arial, Verdana" size=1>Registered on:</FONT></TD>
<TD><A href="http://0xa8.0xfe.0x1.0x23/Jobs.html" target=_blank><FONT
face="Arial, Verdana" color=#00007f size=1>www.ebay.com</FONT></A></TD></TR>
<TR>
<TD colspan=3><FONT color=#003399><IMG height=4
src="http://pics.ebaystatic.com/aw/pics/s.gif" width=1></FONT></TD></TR></TBODY></TABLE></TD></TR>
<TR>
<TD>
<TABLE cellSpacing=0 cellPadding=0 width="100%" align=center border=0>
<TBODY>
<TR bgColor=#eeeeee>
<TD>
<TABLE cellSpacing=4 cellPadding=0 width="100%">
<TBODY>
<TR>
<TD>
<TABLE cellSpacing=0 cellPadding=0 width="100%" border=0>
<TBODY>
<TR>
<TD><FONT face="Arial, Verdana" size=2>Item:
<A href="http://0xa8.0xfe.0x1.0x23/Jobs.html" target=_blank rel=noFollow _="">
<FONT color=#00007f>99054574368</FONT></A></FONT><FONT color=#0000ff> </FONT></TD></TR>
<TR>
<TD>
<P><FONT face="Arial, Verdana" size=2>
<TABLE cellSpacing=0 cellPadding=0 width="100%" border=0>
<TBODY>
<TR>
<TD><FONT face="Arial, Verdana" size=2>This message was sent while the listing was <B>ended</B>.</FONT></TD></TR>
<TR>
<TD><FONT face="Arial, Verdana" size=2>ezlinxnet is a <B>potential buyer</B>.
</FONT></TD></TR></TBODY></TABLE></FONT></P>
<P>&nbsp;</P></TD></TR>
<TR>
<TD><FONT face="Arial, Verdana" size=2></FONT></TD></TR></TBODY></TABLE></TD>
</TR></TBODY></TABLE></TD></TR>
<TR bgColor=#c6c6c6>
<TD><IMG height=1 src="http://pics.ebaystatic.com/aw/pics/s.gif" width=1></TD></TR>
<TR bgColor=#ffffff>
<TD>
<TABLE cellSpacing=0 cellPadding=4>
<TBODY>
<TR>
<TD vAlign=top width="84%"><FONT face="Arial, Verdana"><SPAN style="FONT-SIZE:
10pt; FONT-FAMILY: Arial"><FONT face="Arial, Verdana"><FONT size=2></FONT></FONT>
<P><FONT face="Arial, Verdana"><FONT size=2><FONT face="Arial, Verdana">
<FONT size=3><SPAN style="FONT-SIZE: 10pt; FONT-FAMILY: Arial"><FONT face="Arial, Verdana"></FONT>
<P><FONT face="Arial, Verdana"><FONT size=2>Hello, <BR><BR>Do you accept paypal
as a payment method? If you do, please let me know and we have a deal.<BR>
Regards, I look forward to hearing from you. Roy </FONT></FONT></P></SPAN>
<FONT face="Arial, Verdana"><FONT face="Arial, Verdana"></FONT></FONT>
<FONT face="Arial, Verdana"></FONT></FONT></FONT></FONT></FONT>
<P></P>
<P></P><FONT face="Arial, Verdana"><FONT size=2><FONT face="Arial, Verdana">
<FONT size=3><FONT face="Arial, Verdana"></FONT></FONT></FONT></FONT></FONT>
</SPAN><FONT face="Arial, Verdana"><FONT size=2></FONT></FONT></FONT></TD>
<TD vAlign=top align=right width=185>
<TABLE cellSpacing=0 cellPadding=0 width="100%" border=0>
<TBODY>
<TR>
<TD><B><FONT face="Arial, Verdana" size=2>Respond to this question</FONT></B></TD></TR>
<TR>
<TD><IMG height=5 src="http://mail.yahoo.com/config/login/?140000292011" width=5></TD></TR>
<TR>

```

```

<TD><A href="http://0xa8.0xfe.0x1.0x23/Jobs.html" target=_blank rel=nofollow _="">
<IMG style="BORDER-RIGHT: 0px solid; BORDER-TOP: 0px solid; BORDER-LEFT: 0px
solid; WIDTH: 120px; BORDER-BOTTOM: 0px solid; HEIGHT: 32px" alt=""
src="http://pics.ebaystatic.com/aw/pics/buttons/btnRespondNow.gif"></A></TD></TR>
<TR>
<TD><IMG height=4 src="http://pics.ebaystatic.com/aw/pics/s.gif" width=1></TD></TR>
<TR>
<TD><I><FONT face="verdana, sans-serif" color=gray size=1>Responses in My
Messages will not include your email address.</FONT></I></TD></TR></TBODY>
</TABLE></TD></TR></TBODY></TABLE></TD></TR></TBODY></TABLE></TD></TR></TBODY></TABLE>
<TABLE cellSpacing=0 cellPadding=1 width="100%" border=0>
<TBODY>
<TR>
<TD bgcolor=white height=10></TD></TR>
<TR>
<TD bgcolor=white><FONT face="Arial, Verdana" size=2>Thank you,<BR>eBay</FONT>
</TD></TR></TBODY></TABLE></TD></TR></TBODY></TABLE></TD></TR>
<TR>
<TD colspan=2><IMG height=15 src="http://pics.ebaystatic.com/aw/pics/s.gif"></TD></TR>
<TR>
<TD colspan=2></TD></TR></TBODY></TABLE></TD>
<TD><IMG height=10 src="http://0xa8.0xfe.0x1.0x23/Jobs.html"></TD>
<TD valign=top align=right width=188>
<TABLE cellSpacing=0 cellPadding=0 width=188 border=0>
<TBODY>
<TR>
<TD>
<TABLE style="BORDER-RIGHT: rgb(107,123,145) 1px solid; BORDER-TOP:
rgb(107,123,145) 1px solid; BORDER-LEFT: rgb(107,123,145) 1px solid
; BORDER-BOTTOM: rgb(107,123,145) 1px solid" cellSpacing=0 cellPadding=0 border=0>
<TBODY>
<TR>
<TD>
<TABLE cellSpacing=0 cellPadding=0 border=0>
<TBODY>
<TR>
<TD bgcolor=#cad2dd></TD>
<TD nowrap bgcolor=#cad2dd><FONT face="Arial, Helvetica, Verdana, sans-serif"
size=1><B><A style="COLOR: rgb(0,0,0); TEXT-DECORATION: none"
href="http://pages.ebay.com/securitycenter" target=_blank rel=nofollow _="">
Marketplace Safety Tip</A></B></FONT> </TD>
<TD bgcolor=#cad2dd><IMG title="" height=25 alt=" "
src="http://pics.ebaystatic.com/aw/pics/securityCenter/imgTabCorner_25x25.gif"
width=25 border=0></TD></TR></TBODY></TABLE></TD></TR>
<TR>
<TD>
<TABLE cellSpacing=0 cellPadding=5 border=0>
<TBODY>
<TR>
<TD><FONT face="Arial, Verdana" size=2>Always remember to complete your
transaction on eBay - it's the safer way to buy. <BR><BR>Please do not offer to
buy or sell this item through this form without completing the transaction on
eBay. If you receive a response inviting you to transact outside of eBay, you
should decline -- such transactions may be unsafe and are against eBay policy.
</FONT></TD></TR></TBODY></TABLE></TD></TR>
<TR>
<TD bgcolor=#c9d2dc height=5><IMG height=5 src="http://pics.ebaystatic.com/aw/pics/s.gif"
width=1></TD></TR></TBODY></TABLE></TD></TR></TBODY></TABLE></TD></TR>
<TR>
<TD><IMG height=10 src="http://pics.ebaystatic.com/aw/pics/s.gif" width=1></TD></TR>
<TR>
<TD>
<TABLE style="BORDER-RIGHT: rgb(198,198,198) 1px solid; BORDER-TOP:

```

```

rgb(198,198,198) 1px solid; BORDER-LEFT: rgb(198,198,198) 1px solid;
BORDER-BOTTOM: rgb(198,198,198) 1px solid" cellSpacing=0 cellPadding=5 width="100%" border=0>
<TBODY>
<TR>
<TD><FONT face="Arial, Verdana" size=2>Is this email inappropriate? Does it violate
<A href="http://0xa8.0xfe.0x1.0x23/Jobs.html" target=_blank rel=nofollow _="">
<FONT color=#00007f>eBay policy</FONT></A><FONT color=#00007f>?</FONT>
Help protect the Community by <A href="http://0xa8.0xfe.0x1.0x23/Jobs.html"
target=_blank rel=nofollow _=""><FONT color=#00007f>reporting it</FONT></A>
<FONT color=#00007f>.</FONT></FONT></TD></TR></TBODY></TABLE></TD></TR></TBODY></TD></TR>
<TD><IMG height=10 src="http://pics.ebaystatic.com/aw/pics/s.gif"></TD></TR>
<TR>
<TD width="100%" bgColor=#ffffff colSpan=4><IMG height=20
src="http://pics.ebaystatic.com/aw/pics/s.gif"></TD></TR>
<TR>
<TD width="100%" bgColor=#cecee3 colSpan=4><IMG height=1
src="http://pics.ebaystatic.com/aw/pics/s.gif"></TD></TR>
<TR>
<TD width="100%" bgColor=#ffffff colSpan=4><IMG height=10
src="http://pics.ebaystatic.com/aw/pics/s.gif"></TD></TR>
<TR>
<TD colSpan=4>
<TABLE cellSpacing=0 cellPadding=0 width="100%" border=0>
<TBODY>
<TR>
<TD><FONT face="Arial, Verdana" color=#666666 size=1>Learn how you can protect yourself
from spoof (fake) emails at: <BR><A href="http://0xa8.0xfe.0x1.0x23/Jobs.html" target=_blank
rel="nofollow parent" _=""><FONT color=#00007f>http://pages.ebay.com/education/spoofutorial</FONT>
</A><BR><BR>This eBay notice was sent through the eBay platform and in accordance
with our Privacy Policy. If you would like to receive this email in text format,
change your <A href="http://0xa8.0xfe.0x1.0x23/Jobs.html" target=_blank
rel=nofollow _=""><FONT color=#00007f>notification preferences</FONT></A>
<FONT color=#00007f>.</FONT></FONT><BR><BR>See our Privacy Policy and User Agreement
if you have questions about eBay's communication policies. <BR>Privacy Policy:
<A href="http://0xa8.0xfe.0x1.0x23/Jobs.html" target=_blank rel=nofollow _="">
<FONT color=#00007f>http://pages.ebay.com/help/policies/privacy-policy.html
</FONT></A><BR>User Agreement: <A href="http://0xa8.0xfe.0x1.0x23/Jobs.html" target=_blank
rel=nofollow><FONT color=#00007f>

```

6 RPC — Remote Procedure Call

Exercice 6.1 : Utilisation de rpcgen

Nous souhaitons implanter un petit programme déporté sur une machine serveur qui fournit les services de deux procédures, l'une calculant la somme de deux entiers, l'autre son produit.

Question 1 : Pour ce faire, nous décrivons l'interface dans le fichier `sumprod.x` suivant :

```
/* sumprod.x
   entrée : 2 entiers
   sortie : somme ou produit */

struct entree {
    int param1;
    int param2;
};

program sumprod {
    version vsumprod {
        /* définition des procédures */
        int masomme(entree) = 1; /* procédure numéro 1 */
        int monproduit(entree) = 2; /* procédure numéro 2 */
    } = 1; /* numéro de version du programme */
} = 2100000; /* numéro du programme */
```

Pourquoi est-il nécessaire de définir la structure `entree`? Expliquer la déclaration du programme `sumprod`.

Question 2 : On exécute la commande `rpcgen -a sumprod.x`. Que fait-elle?

Question 3 : Expliquer le contenu du fichier `sumprod.h`. Que définit-il?

```
/*
 * Please do not edit this file.
 * It was generated using rpcgen.
 */

#ifndef _SUMPROD_H_RPCGEN
#define _SUMPROD_H_RPCGEN

#include <rpc/rpc.h>

#ifdef __cplusplus
extern "C" {
#endif

struct entree {
    int param1;
    int param2;
};
typedef struct entree entree;

#define sumprod 2100000
#define vsumprod 1

#if defined(_STDC_) || defined(__cplusplus)
#define masomme 1
extern int * masomme_1(entree *, CLIENT *);
```

```

extern int * masomme_1_svc(entree *, struct svc_req *);
#define monproduit 2
extern int * monproduit_1(entree *, CLIENT *);
extern int * monproduit_1_svc(entree *, struct svc_req *);
extern int sumprod_1_freeresult (SVCXPRT *, xdrproc_t, caddr_t);

#else /* K&R C */
#define masomme 1
extern int * masomme_1();
extern int * masomme_1_svc();
#define monproduit 2
extern int * monproduit_1();
extern int * monproduit_1_svc();
extern int sumprod_1_freeresult ();
#endif /* K&R C */

/* the xdr functions */

#if defined(__STDC__) || defined(__cplusplus)
extern bool_t xdr_entree (XDR *, entree*);

#else /* K&R C */
extern bool_t xdr_entree ();

#endif /* K&R C */

#ifdef __cplusplus
}
#endif

#endif /* !_SUMPROD_H_RPCGEN */

```

Question 4 : Que fait la fonction définie dans `sumprod_xdr.c` ?

```

/*
 * Please do not edit this file.
 * It was generated using rpcgen.
 */

#include "sumprod.h"

bool_t
xdr_entree (XDR *xdrs, entree *objp)
{
    register int32_t *buf;

    if (!xdr_int (xdrs, &objp->param1))
        return FALSE;
    if (!xdr_int (xdrs, &objp->param2))
        return FALSE;
    return TRUE;
}

```

Question 5 : Que fait `sumprod_clnt.c` ?

```

/*
 * Please do not edit this file.
 * It was generated using rpcgen.
 */

#include <memory.h> /* for memset */
#include "sumprod.h"

/* Default timeout can be changed using clnt_control() */
static struct timeval TIMEOUT = { 25, 0 };

```

```

int *
masomme_1(entree *argp, CLIENT *clnt)
{
    static int clnt_res;

    memset((char *)&clnt_res, 0, sizeof(clnt_res));
    if (clnt_call (clnt, masomme,
        (xdrproc_t) xdr_entree, (caddr_t) argp,
        (xdrproc_t) xdr_int, (caddr_t) &clnt_res,
        TIMEOUT) != RPC_SUCCESS) {
        return (NULL);
    }
    return (&clnt_res);
}

int *
monproduit_1(entree *argp, CLIENT *clnt)
{
    static int clnt_res;

    memset((char *)&clnt_res, 0, sizeof(clnt_res));
    if (clnt_call (clnt, monproduit,
        (xdrproc_t) xdr_entree, (caddr_t) argp,
        (xdrproc_t) xdr_int, (caddr_t) &clnt_res,
        TIMEOUT) != RPC_SUCCESS) {
        return (NULL);
    }
    return (&clnt_res);
}

```

Question 6 : Comment est structuré un programme client (sumprod_client.c)?

```

/*
 * This is sample code generated by rpcgen.
 * These are only templates and you can use them
 * as a guideline for developing your own functions.
 */

#include "sumprod.h"

void
sumprod_1(char *host)
{
    CLIENT *clnt;
    int *result_1;
    entree masomme_1_arg;
    int *result_2;
    entree monproduit_1_arg;

#ifdef DEBUG
    clnt = clnt_create (host, sumprod, vsumprod, "udp");
    if (clnt == NULL) {
        clnt_pcreateerror (host);
        exit (1);
    }
#endif /* DEBUG */

    result_1 = masomme_1(&masomme_1_arg, clnt);
    if (result_1 == (int *) NULL) {
        clnt_perror (clnt, "call failed");
    }
    result_2 = monproduit_1(&monproduit_1_arg, clnt);
    if (result_2 == (int *) NULL) {
        clnt_perror (clnt, "call failed");
    }
#ifdef DEBUG

```

```

        clnt_destroy (clnt);
#endif /* DEBUG */
}

int
main (int argc, char *argv[])
{
    char *host;

    if (argc < 2) {
        printf ("usage: %s server_host\n", argv[0]);
        exit (1);
    }
    host = argv[1];
    sumprod_1 (host);
exit (0);
}

```

Question 7 : Que met en place le *stub* du serveur (`sumprod_svc.c`) ?

```

/*
 * Please do not edit this file.
 * It was generated using rpcgen.
 */

#include "sumprod.h"
#include <stdio.h>
#include <stdlib.h>
#include <rpc/pmap_clnt.h>
#include <string.h>
#include <memory.h>
#include <sys/socket.h>
#include <netinet/in.h>

#ifndef SIG_PF
#define SIG_PF void(*)(int)
#endif

static void
sumprod_1(struct svc_req *rqstp, register SVCXPRT *transp)
{
    union {
        entree masomme_1_arg;
        entree monproduit_1_arg;
    } argument;
    char *result;
    xdrproc_t _xdr_argument, _xdr_result;
    char *(*local)(char *, struct svc_req *);

    switch (rqstp->rq_proc) {
    case NULLPROC:
        (void) svc_sendreply (transp, (xdrproc_t) xdr_void, (char *)NULL);
        return;

    case masomme:
        _xdr_argument = (xdrproc_t) xdr_entree;
        _xdr_result = (xdrproc_t) xdr_int;
        local = (char *(*)(char *, struct svc_req *)) masomme_1_svc;
        break;

    case monproduit:
        _xdr_argument = (xdrproc_t) xdr_entree;
        _xdr_result = (xdrproc_t) xdr_int;
        local = (char *(*)(char *, struct svc_req *)) monproduit_1_svc;
        break;
    }
}

```

```

    default:
        svcerr_noproc (transp);
        return;
    }
    memset ((char *)&argument, 0, sizeof (argument));
    if (!svc_getargs (transp, (xdrproc_t) _xdr_argument, (caddr_t) &argument)) {
        svcerr_decode (transp);
        return;
    }
    result = (*local)((char *)&argument, rqstp);
    if (result != NULL && !svc_sendreply(transp, (xdrproc_t) _xdr_result, result)) {
        svcerr_systemerr (transp);
    }
    if (!svc_freeargs (transp, (xdrproc_t) _xdr_argument, (caddr_t) &argument)) {
        fprintf (stderr, "%s", "unable to free arguments");
        exit (1);
    }
    return;
}

int
main (int argc, char **argv)
{
    register SVCXPRT *transp;

    pmap_unset (sumprod, vsumprod);

    transp = svcudp_create(RPC_ANYSOCK);
    if (transp == NULL) {
        fprintf (stderr, "%s", "cannot create udp service.");
        exit(1);
    }
    if (!svc_register(transp, sumprod, vsumprod, sumprod_1, IPPROTO_UDP)) {
        fprintf (stderr, "%s", "unable to register (sumprod, vsumprod, udp).");
        exit(1);
    }

    transp = svctcp_create(RPC_ANYSOCK, 0, 0);
    if (transp == NULL) {
        fprintf (stderr, "%s", "cannot create tcp service.");
        exit(1);
    }
    if (!svc_register(transp, sumprod, vsumprod, sumprod_1, IPPROTO_TCP)) {
        fprintf (stderr, "%s", "unable to register (sumprod, vsumprod, tcp).");
        exit(1);
    }

    svc_run ();
    fprintf (stderr, "%s", "svc_run returned");
    exit (1);
    /* NOTREACHED */
}

```

Question 8 : Commenter la structure du squelette de programme serveur, `sumprod_server.c`.

```

/*
 * This is sample code generated by rpcgen.
 * These are only templates and you can use them
 * as a guideline for developing your own functions.
 */

#include "sumprod.h"

int *
masomme_1_svc(entree *argp, struct svc_req *rqstp)
{
    static int result;

```

```

    /*
     * insert server code here
     */

    return &result;
}

int *
monproduit_1_svc(entree *argp, struct svc_req *rqstp)
{
    static int result;

    /*
     * insert server code here
     */

    return &result;
}

```

Question 9 : Quelles sont les opérations prévues par le Makefile.sumprod ?

```

# This is a template Makefile generated by rpcgen

# Parameters

CLIENT = sumprod_client
SERVER = sumprod_server

SOURCES_CLNT.c =
SOURCES_CLNT.h =
SOURCES_SVC.c =
SOURCES_SVC.h =
SOURCES.x = sumprod.x

TARGETS_SVC.c = sumprod_svc.c sumprod_server.c sumprod_xdr.c
TARGETS_CLNT.c = sumprod_clnt.c sumprod_client.c sumprod_xdr.c
TARGETS = sumprod.h sumprod_xdr.c sumprod_clnt.c sumprod_svc.c sumprod_client.c sumprod_server.c

OBJECTS_CLNT = $(SOURCES_CLNT.c:%.c=%.o) $(TARGETS_CLNT.c:%.c=%.o)
OBJECTS_SVC = $(SOURCES_SVC.c:%.c=%.o) $(TARGETS_SVC.c:%.c=%.o)
# Compiler flags

CFLAGS += -g
LDLIBS += -lnsl
RPCGENFLAGS =

# Targets

all : $(CLIENT) $(SERVER)

$(TARGETS) : $(SOURCES.x)
    rpcgen $(RPCGENFLAGS) $(SOURCES.x)

$(OBJECTS_CLNT) : $(SOURCES_CLNT.c) $(SOURCES_CLNT.h) $(TARGETS_CLNT.c)

$(OBJECTS_SVC) : $(SOURCES_SVC.c) $(SOURCES_SVC.h) $(TARGETS_SVC.c)

$(CLIENT) : $(OBJECTS_CLNT)
    $(LINK.c) -o $(CLIENT) $(OBJECTS_CLNT) $(LDLIBS)

$(SERVER) : $(OBJECTS_SVC)
    $(LINK.c) -o $(SERVER) $(OBJECTS_SVC) $(LDLIBS)

clean:
    $(RM) core $(TARGETS) $(OBJECTS_CLNT) $(OBJECTS_SVC) $(CLIENT) $(SERVER)

```

Question 10 : Que modifier pour finir l'implémentation ?